Attack Pattern Glossary

Sean Barnum, Cigital, Inc. [vita[3]]
Amit Sethi, Cigital, Inc. [vita[4]]

2006-11-07

The Attack Pattern articles use the following terms and definitions.

attack	An attack is the act of carrying out an exploit.
attack path	An attack path is a path in an attack tree from a leaf node to the root node. An attack path can be a simplistic representation of an attack pattern.
attack pattern	An attack pattern is a general framework for carrying out a particular type of attack such as a particular method for exploiting a buffer overflow or an interposition attack that leverages architectural weaknesses. In this paper, an attack pattern describes the approach used by attackers to generate an exploit against software.
attack tree	Attack trees (known as “threat trees” by Microsoft) provide a formal, methodical way of describing the security of systems based on various attacks [Schneier[6] 99[7]]. The root node of the tree is the attacker’s goal (known as “threat” by Microsoft), and the “children” of each node describe a lower-level way of achieving the goal of the parent node. In this manner, the leaf nodes generally contain relatively low-level tasks such as “install a key logger on target machine”, and the root node contains a goal such as “obtain administrator’s password.”
attacker	An attacker is the person that actually executes an attack. Attackers may range from very unskilled individuals leveraging automated attacks developed by others (“script kiddies”) to well-funded government agencies or even large international organized crime syndicates with highly skilled software experts.
bugs	Bugs are software problems that exist only in code. A bug that exists in code may or may not ever be executed or exploitable. Therefore, a bug may or may not represent a vulnerability in the underlying software. Bugs are used to describe minor implementation errors that are typically easy to fix. Note that simply because bugs are minor implementation errors does not mean that the impact of an attacker exploiting the bug is small. For

3. http://buildsecurityin.us-cert.gov/bsi/about_us/authors/35-BSI.html (Barnum, Sean)
4. http://buildsecurityin.us-cert.gov/bsi/about_us/authors/601-BSI.html (Sethi, Amit)

	instance, a buffer overflow is a well-known type of bug that is generally easy to fix. However, exploiting a buffer overflow can give an attacker full control over a system.
design pattern	A design pattern is a general repeatable solution to a recurring software engineering problem.
exploit	An exploit is a technique or software code (often in the form of scripts) that takes advantage of a vulnerability[8] or security weakness in a piece of target software.
flaws	Flaws are software problems that exist in the software's design. A flaw may or may not represent a vulnerability in the underlying software. Mitigating a flaw typically involves significantly more effort than simply modifying a few lines of code. The problem does not lie solely in the implementation; the underlying design is flawed, and therefore, any implementation that follows the design would contain the flaw. For instance, performing sensitive business logic in an untrusted client application is a design flaw that cannot be mitigated by a simple measure such as modifying array bounds.
impact	Impact is the effect that the organization using vulnerable software faces if a vulnerability were to be exploited. Impact could range from specific tangible values such as monetary fines from the breach of a law or regulation to intangible values such as brand and reputation damage.
misuse/abuse cases	Misuse/Abuse cases can be viewed as use case requirements with an attacker as the actor. They represent actions taken against the software that do not fall within the normal, defined operating parameters of the system. They are typically identified and modeled in an integrated fashion with positive functional use cases for a system.
risk	Risks capture the likelihood that a vulnerability will be exploited, as well as the potential damage (impact) that will occur if it is. It is important to note that risks, threats, and exploits are all separate things. Risks may be present in the target software, on the target host, or in the broader operational environment of the software.
risk analysis	Risk analysis involves analyzing target software for vulnerabilities and characterizing their nature and potential impact. Microsoft calls this "threat modeling." Risk analysis attempts to identify, prioritize, and plan appropriate mitigation for the risks facing a piece of software.
security pattern	A security pattern is a design pattern that is intended to show software developers how to design and

	implement solutions to common security problems. These solutions typically represent software security features. A security pattern may be used to mitigate multiple attack patterns, and an attack pattern may be mitigated using multiple security patterns.
target host	A target host is the computer or platform that is running the target software of an attack. A host may be attacked through interfaces of the target software or through purely network-based attack mechanisms.
target software	Target software is software that is the target of an attack.
threat	A threat is an actor or an agent that is a source of danger to the system under consideration or the assets to which it has access. The threat can be a person that abuses the software, a program running on a compromised system, or even a non-sentient event such as a hardware failure. A threat exploits a vulnerability in software to attack it.
vulnerabilities	A vulnerability is a software weakness that can be exploited by an attacker. Bugs and flaws collectively form the basis of most software vulnerabilities.
weakness	A weakness is an underlying condition or construct existing in a software system that has the potential for negatively impacting the security of the system.

Cigital, Inc. Copyright

The Build Security In (BSI) portal is sponsored by the U.S. Department of Homeland Security (DHS), National Cyber Security Division. The Software Engineering Institute (SEI) develops and operates BSI. DHS funding supports the publishing of all site content.

1. mailto:copyright@cigital.com